

THE SEYBOLD CONSULTING GROUP, INC.

John W. Seybold
Jonathan W. Seybold
Patricia B. Seybold
Joseph L. Ehardt

5 October 1982

A Review of the Datapoint User Interface Language,
Vantage, and the Electronic Mail User Interface

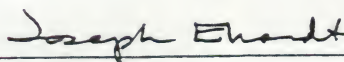
TECHNICAL PLANNING AND ASSISTANCE

Prepared for:

Datapoint Corporation
9725 Datapoint Drive
San Antonio, Texas 78284

Attention of: Mr. David Ruberg
Vice President
Software Products Division

Submitted by:



Joseph Ehardt, Senior Consultant,
The Seybold Consulting Group, Inc.

A REVIEW OF THE DATAPoint USER INTERFACE LANGUAGE,
VANTAGE, AND THE ELECTRONIC MAIL USER INTERFACE

This report comments upon various Datapoint software systems to be released in the near future. These include the User Interface Language (UIL), Vantage, and the user interface of the Electronic Mail System. Naturally, the contents of this report are strictly confidential and may not be used by Datapoint as a public endorsement of ~~the~~ its efforts by the Seybold companies or their affiliates.

I Miller's Columns

For many years we have advocated the use of multi-level file directories for storing documents, archived messages, data files, and the like. In our February 1980 issue, we took a public position on this matter, saying that disks of increasing capacity make the need for such filing structures obvious.

When the Bell Laboratories' UNIX system was implemented for minicomputer systems, such multi-level directories were provided, but the notation for accessing directories and the files contained therein was very stylistic. A technical user familiar with data processing systems felt it was perfectly acceptable, but many office personnel felt uncomfortable with it because it was easy for them to forget the level of the filing structure they were in. In other words, the technical structure of directories containing sub-directories and other files was fine, but the human interface was less than desirable for comfortable use in the general office setting.

So when we saw the Datapoint implementation of a three-column presentation (called Miller's Columns) showing the higher level, current level, and lower level of the directory structure, it was clear that our desire for a more visual portrayal of the filing structure had been satisfied. We like the implementation very much, though there is one detraction which bears mentioning.

The "active" item is always displayed in the center of the screen (it is highlighted using inverse video) to make it obvious that it is the current selection. No matter what column you look at, the highlighted item is the repository of the items in the column to the right. The

UP and DOWN cursor keys are used to scroll the center column vertically, and the LEFT and RIGHT cursor keys are used to go higher or lower in the filing structure, which is to say to the left or right column on the screen.

The problem is that the cursor keys are used as scrolling keys. Pressing the UP or DOWN cursor key causes the lines in the center column to vertically scroll while the highlighting bar remains in the center of the screen. When you stop, the column to the right shows the contents of the center item. With most systems, these keys cause the cursor to move up and down rather than cause the information to scroll, so we were completely unprepared for the system's behavior.

Ironically, it is the Datapoint keyboard which continues to be the constricting factor for many of its software subsystems. The cursor keys, which are clustered in a nice pattern to permit movement in all four directions, are the most logical group of keys to use in spite of the break from convention. Still, to us as well as many others, cursor keys are very different from scrolling keys, and the implementation of Miller's Columns interchanges the two as if they are synonymous. We can only hope that the future Datapoint keyboard is less of a problem.

To return to a positive note, we liked the capability whereby the user can employ any short phrase as an item within a column rather than be restricted to the stylized file names that computers often require. This leads us to a discussion of the user interface language.

II User Interface Language (UIL)

We find the user interface language to be much more oriented toward the general office environment than is normal for computers systems. We especially like the way the user is shown the various alternatives that may be chosen.

Whereas some solutions are designed from the very beginning to be easy to use and understand, which is the approach taken by the Xerox Star professional workstation, more traditional systems are greatly confined by the multitude of programs which have been written without significant attention being paid to this issue.

The UIL, on the other hand, is designed to present the user with the various options of an existing program in a more humane way and then communicate the necessary information to that program in the typically stylized manner. Thus it performs a unique service by making old programs easier to use. We think the marketplace will react very

favorably to the UIL, and we hope that it stimulates many programmers to be more conscious of interface issues since the effort needed to make programs more pleasant has been greatly lessened.

III Vantage

This enhancement is designed to provide the user with a convenient way to switch from one task to another with the simple tap of a key or two. In the early implementation, each task controls a virtual screen of the same dimensions as the physical screen. Vantage simply allows a given virtual screen to be displayed on the CRT with ease. In a future release, each virtual screen can be viewed through a window, and the CRT may be showing a number of windows at the same time.

Here again is evidence that Datapoint is making old programs function in an environment that is much kinder to the users than before. For example, with Vantage, the user can readily switch from the mail system to Multiplan, refer to several figures on a spreadsheet, and then return to the mail system to complete the message. We have worked with systems with similar facilities, though none are quite as nice as the Vantage implementation, and have found them invaluable.

From a competitive aspect, we think that Datapoint must have a Vantage-like facility. To have it means that Datapoint is doing everything possible to keep pace with others while maintaining the viability of its existing library of programs. It is obvious that newer systems are less encumbered by existing software libraries, making it ~~X~~ easier to implement a user interface reflecting modern requirements. Without Vantage, Datapoint would begin to fall behind, making its systems less attractive to customers and dealers alike.

IV Mail Interface Facility (MIF)

The RMS electronic mail system is comprised of the mail interface facility (MIF) and the mail movement facility (MMF). We were asked to review the MIF as our principal assignment during this visit.

A. Mail System Files

The electronic mail system stores information in a number of different files:

- The Mailbox File contains the header information of every message received by a user as well as a pointer to the message body.
- The Message Body File contains the body or textual portion of a message, which remains in the possession of the original sender.
- The User Directory contains a complete list of mail system users.
- The Profile File contains all information defined by the user regarding message directory formatting, the various canned phrases that identify the message header fields, as well as some other useful information.
- The Distribution List File contains a list of names for recurrent distribution of messages.

From the descriptions of these files, it is apparent that Datapoint is providing the framework for a mail system of considerable flexibility. We have some suggestions for improvements, but we will hold these comments until we get to the appropriate place in our review.

B. Entering the Mail System

When the user starts the mail system, the command can include several optional arguments. These include the name of the mailbox to be accessed, the name of the profile file, and an alias to identify the agent (such as one's secretary) who is entering the message.

After entry, one can process incoming mail contained in the mailbox, search through the mailbox for items matching certain criteria, send outgoing mail, run some utility functions (e.g., editing a profile), and exit the mail system. These are displayed as commands at the bottom of the screen, and you can select the desired command by pressing the RIGHT or LEFT cursor key until the word is highlighted, or by typing its initial letter. Datapoint has organized the command list according to frequency of use, with the default being to process incoming mail.

Although many mail systems will immediately show the user any unread mail items, the Datapoint method requires only a tap of the RETURN key to confirm that incoming mail is to be handled. Thus the minimal amount of extra effort can be ignored. But we were disappointed that the system did not take the opportunity to provide statistics so that the user could most effectively use their time. For example, the system might show us:

The MAILBOX contains	
148 messages, including	48 messages delivered
<u>23 unread urgent messages</u>	<u>4 messages rejected</u>
3 unread messages	5 messages in transit
122 messages already read	1 confirmation

With the current implementation, the user does not know beforehand what exists in the mailbox (even unread messages!) since there are no revealing statistics. So the user must ask the system to find each situation with separate lookups, each of which consumes system cycles even though the outcome is uncertain. With statistics, the user can decide whether to immediately look for the rejected messages, for the urgent messages, or for the confirmed message, as they see fit, all before handing the "normal" incoming messages.

Many equally useful ideas are compatible with the basic approach taken by Datapoint, but this example conveys the basis of our argument. And since we consistently advocate that the computer make life easier for the user, we urge that Datapoint consider providing statistics.

And finally, we would like to return to an issue we briefly touched upon before. We question the wisdom of trying to organize all command lists according to some arbitrary determination of frequency of use. The same command (e.g., SEND) will appear in completely different locations within the command list depending upon the command context and the presumed frequency. We suggest that the command list be displayed in alphabetical order so that the user can verify the legality of a given command by looking beforehand. Furthermore, the "most frequent" command can immediately be highlighted for immediate execution, no matter where it is located within the command list, though the user is free to select any command they wish.

C. Processing Incoming Mail

When the user asks for incoming mail, the system shows a directory of message headers according to the format specified in the profile file. Typical information would include who the message was from, when it was sent, its size in terms of 256-character blocks, and its subject.

Message headers are shown in inverse chronological order, which is to say that the most recent message is listed first, and so on. We prefer to allow the user to control the order in which messages are listed. Since this sort facility is rarely provided, the most useful sequence is to show message headers in normal chronological order. In this way, the oldest message is emphasized by being first, and the user is encouraged to read it immediately. In the same vein, if two or more messages have been received from someone regarding the same subject, it clearly is better to read these messages in the order in which they were sent. After all, we do not want our telephone calls to come in reverse order, so why should printed messages be handled differently?

During our visit it was mentioned that message size is shown in terms of the number of 256-character blocks needed to store the message. Such feedback of length is virtually useless to anyone except programmers, so Datapoint might have simply classified messages as very short, short, medium, long and very long. In fact, we suggest that the message length be shown in terms of the number of text lines in the message body.

D. Reading a Message

Given a list of message headers, you can select one with the cursor and read it, forward it to someone else, print it, or delete it. When you read it, the screen is filled with the message, both the header in the prescribed form and the text body.

Generally speaking, there is no problem with the way in which Datapoint proposes to handle this. However, when the number of names in the "TO:" or "COPY TO:" list is greater than eight, we suggest replacing the "...." as an indication of more unlisted names into a message saying "(and 4 others)" so that the user knows the precise number of omitted names.

Unfortunately, scrolling through the message is somewhat contrived because of the tricky way in which the mail system uses the KBD and DSP keys, but this is induced primarily by the Datapoint keyboard, a problem that we have mentioned elsewhere.

A point which interested us is the message body is stored in the file space of the original sender. This is unusual though it does have its benefits. For example, only one copy of the body is needed even if there are many recipients. Other mail systems usually store the message in their own file space and place pointers in the mailboxes of the various recipients. We feel there is bothersome problem with the Datapoint method when users are on different networks, possibly separated physically by great distances. In this case, recipients will be reading messages which are stored remotely rather than locally, thereby increasing communications costs. We feel sure that Datapoint will examine this issue carefully before a final storage method is proposed.

E. Forwarding a Message

When the user forwards a message, several key fields are prepended to the original message. To make the original message clearly stand out from the new header, we suggest introducing a horizontal line of dashes with the embedded phrase "Forwarded Message Follows" to separate the new from the old header.

Furthermore, current plans are to include the original message as text within the new message, a course that appears quite reasonable. But this permits the forwarder to alter the text of the original message, an act which poses serious social and legal problems. Basically, we take the position that one should not be able to change another person's message. If necessary, one person can provide another with a copy of that information, but this is done by an explicit act by the originator. Systems should preserve the ownership of information wherever possible or a backlash will result as a natural course of events.

F. Processing Outgoing Mail

When entering this segment of mail handling, the user is shown a list of all partially written messages in roughly the same format as incoming mail. One can modify any of these messages, send it off, or create another message as needs require. We find this basic approach to be very good, providing the user with considerable flexibility.

G. Creating a Message

The fact that a special message editor is used rather than the standard text editor is very disappointing. Using the same editor, with all of the idiosyncrasies that have been carefully learned, is preferred so that the system provides a consistent interface to the user. The mail system compounds this problem by providing two editors. The first

handles the creation and revision of the message header, while the second handles the same functions for the message body. We understand why Datapoint is providing two editors, but it is ill advised. Flipping from one editor to the other via appropriate commands is unprecedented in electronic mail systems, and we would never encourage anyone to implement it this way.

A feature we like very much is the use of AIM to convert a typed name into a formally known user address. We often chastise other mail systems for not informing the user of unknown addressees until much later, possibly the next day. Immediate feedback before sending the message is a distinct advantage since it eliminates many delivery problems beforehand. This check, however, can be made when the user sends the message, thereby eliminating most of the justification for a special editor for message headers.

H. Looking for Messages

The lookup facility provided in the electronic mail system is very useful. It helps the user find messages sent or received according to practically any criteria imaginable, including the date sent, the name of the originator or recipient, some portion of the subject field, as well as a broad list of system- and user-specified keywords.

And if Datapoint were to implement our recommendation about quantifying various categories of messages, the LOOKUP facility could be extended to fetch these items with less effort than is currently required of the user.

Furthermore, it should be possible to execute the LOOKUP command from various places in the command structure. After all, it is unreasonable to ask the user to go to the highest level in the command tree before they can scan their mailbox, thereby creating an excessive amount of command tree traversal.

V Final Remarks

We think that Datapoint is greatly improving its software with these new efforts. The human interface is definitely enriched by Miller's Columns, the User Interface Language, and Vantage. And as far as new applications are concerned, the electronic mail system is clearly needed to bolster Datapoint's efforts in the integrated use of its systems for data processing, word processing and the electronic distribution of that information among users.

This review intentionally focusses on the major points where improvements are worthwhile. We have purposefully tried to minimize remarks that might be considered "nit-picking" so as not to detract from those comments upon which we place more emphasis.

The mail movement facility is excluded from this review, so we are not in a good position to comment on how well it compares with the mail distribution mechanisms provided by other vendors. Basically we were asked to review only the MIF as opposed to the MMF. Its influence, however, on the operational cost of electronic mail systems is indisputable. From what we gather, Datapoint will provide its customers with an excellent mechanism for distributing information in a cost effective manner, which is the dominant concern.

In conclusion, Datapoint is progressing nicely, and we hope that our comments are listened to with the same care that we take in making them.